Airtable take-home

Thomas Debeauvais, November 2019

I used Modeanalytics.com to query and plot the data. The SQL dialect is Postgres.
https://app.mode.com/editor/gentimouton/reports/a34dd2402031 (may require signing on)

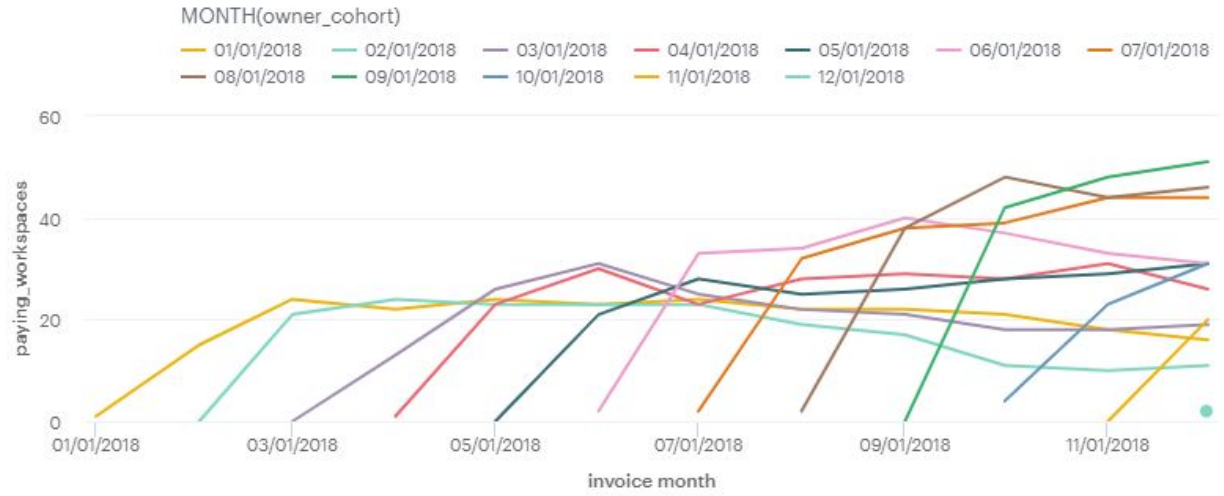
Q1

Instructions: Using this dataset, please calculate and plot recurring revenue over time. Some workspaces may churn once they run out of credits. To account for this, exclude workspaces that have paid less than $10 (excluding credits) by the time the invoice is generated.

Q2

Instructions: Using the first initial invoice month as a cohort, build a cohort analysis table which shows the number of paid seats and number of paying workspaces for each cohort for every subsequent month.

number of paying workspaces monthly, by workspace owner cohort



seats per paying workspace, by owner cohort

Q3

Instructions: Given this dataset, is there any insight that you can glean regarding how this platform has been growing and what sort of growth we can expect in the future?

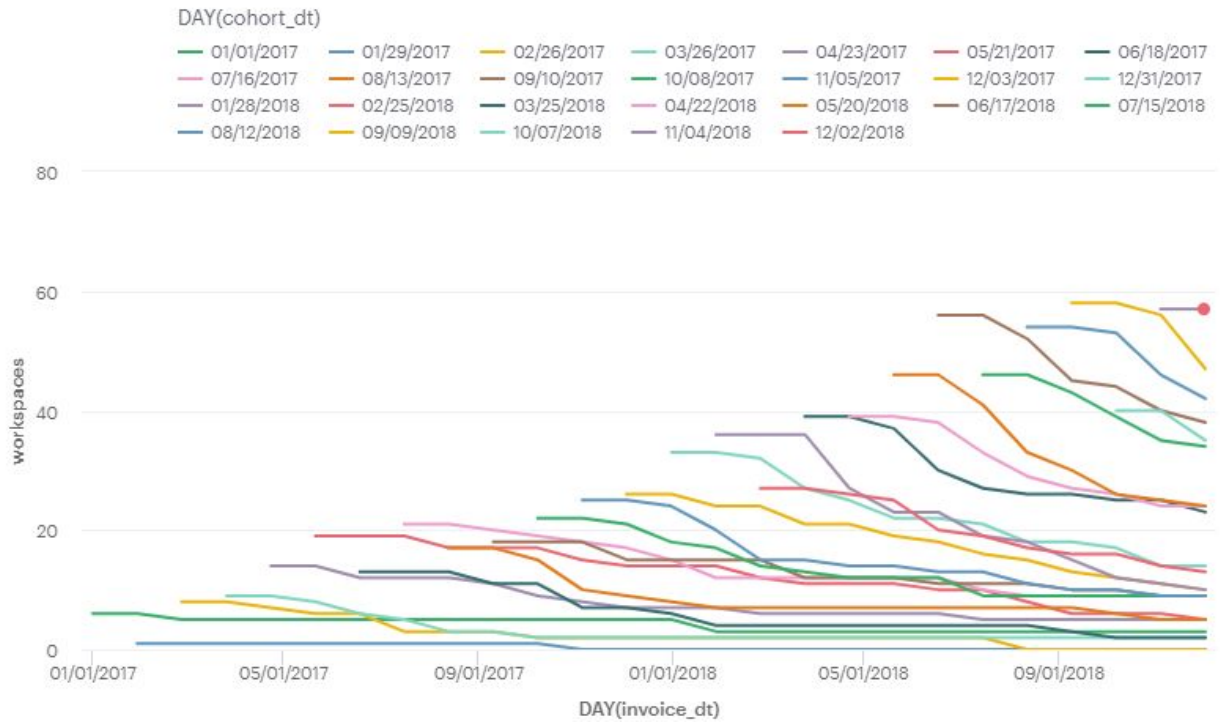
Take-aways:

- **Acquisition is strong: the number of initial invoices is growing every period.**
- **However, retention has not improved over time** (despite probably having released more features). 1-year retention varies drastically between cohorts, ranging between 20 and 60%.

Notes:

- This section uses a 28d definition for invoice period and cohort. This fits the invoice frequency better, by counting each workspace only once per period.

workspaces retained, cohorted by owner's first invoice 28d period



workspace churn against owner age, cohorted (2017 cohorts)

Appendix: queries

Q1

```
-- Using this dataset, please calculate and plot recurring revenue over time.
-- Some workspaces may churn once they run out of credits.
-- To account for this, exclude workspaces that have paid
-- less than $10 (excluding credits) by the time the invoice is generated.

with enriched_invoices as (
select
  *,
  date_trunc('day',date) as dt,
  total_cost_in_cents - total_paid_in_credits_in_cents as revenue
from gentimouton.invoices
)

, to_date_invoices as (
select
  *,
  sum(revenue) over(
          partition by workspace_id order by date
          rows between unbounded preceding and 1 preceding
          ) as revenue_to_date
from enriched_invoices i
)

select
  date_trunc('month',dt) as dt,
  sum(revenue)/100 as revenue_dollars
from to_date_invoices i
where "type" != 'initial'
  and revenue_to_date >= 10
  and dt < '2019-01-01' -- partial invoice period
group by 1
order by 1
```

Q2

```
-- ### Question 2:
-- Using the first initial invoice month as a cohort, build a cohort analysis
-- table which shows the number of paid seats and number of paying workspaces
-- for each cohort for every subsequent month.


with first_workspace_invoice as (
select
  workspace_id,
  min("date") as first_workspace_invoice_ts
from gentimouton.invoices i
where "type" = 'initial'
group by 1
)

, workspace_metadata as (
-- compute workspace owner and owner cohort in the same CTE, to save us a join later
select distinct -- ensure one row per workspace
  workspace_id,
  first_value(owner_id) over( -- in case workspaces had 2+ rows per workspace
          partition by workspace_id order by i.first_workspace_invoice_ts
  ) as owner_id,
  first_value(date_trunc('month',i.first_workspace_invoice_ts)) over( -- this is about owner, not
workspace
          partition by owner_id order by i.first_workspace_invoice_ts
  ) as owner_cohort
from first_workspace_invoice i
join gentimouton.workspaces w -- TODO: double check there's 1 row per workspace
  on w.id = i.workspace_id
)

, workspace_stats_by_invoice_period as (
select
  i.workspace_id as workspace_id,
  i."date" as invoice_ts,
  count(*) as seats, -- TODO: this assumes the owner is NOT a collaborator
  max(total_cost_in_cents-total_paid_in_credits_in_cents) as revenue -- dummy max, to count
revenue only once
```

```
from gentimouton.invoices i
left join gentimouton.workspace_collaborators c
  on c.workspace_id = i.workspace_id
  and date_trunc('day',i."date") >= date_trunc('day',c.created_time)
  and date_trunc('day',i."date") <= coalesce(date_trunc('day',c.deleted_time), current_date)
group by 1,2
)

-- if 2 invoices happen in the same month for a workspace,
-- we sum up the revenue and seats of the 2 invoices together,
-- and also count that workspace twice if both invoices had revenue > 0
-- reasons: 1) we are paid twice, not just once.
-- 2) at the end of the year, we want the numbers to sum up to total revenue.

select
  owner_cohort,
  date_trunc('month',invoice_ts) as dt,
  count(case when revenue>0 then w.workspace_id else null end) as paying_workspaces, --
distinct would count once
  sum(seats) as seats, -- if workspace has 2 invoices this month, it contributes 2 numbers of
seats
  sum(case when revenue>0 then seats else 0 end) ::float / nullif(sum(case when revenue>0
then 1 else 0 end),0) as seats_per_paying_workspace
from workspace_stats_by_invoice_period w
join workspace_metadata m
  on w.workspace_id = m.workspace_id
where invoice_ts < '2019-01-01 00:00:00' -- last and partial invoice period
group by 1,2
order by 1,2
```

Q3

```
-- workspace retention

with enriched_invoices as (
select
  i.workspace_id,
  w.owner_id,
  '2017-01-01'::date + floor(
          (i."date"::date - '2017-01-01'::date)/28
  ) * interval '28 day' as bin_28d
```

```
from gentimouton.invoices i
join gentimouton.workspaces w
  on w.id = i.workspace_id
group by 1,2,3 -- just in case
)

, owner_cohort as (
select
  owner_id,
  min(bin_28d) as owner_cohort_bin_28d
from enriched_invoices
group by 1
)

-- compute all bins so the plot looks nice (lines reach zero)
, all_28d_bins as (
select distinct
  '2017-01-01'::date + floor(
          (i."date"::date - '2017-01-01'::date)/28
  ) * interval '28 day' as bin_28d
from gentimouton.invoices i
)

, all_28d_bins_cohorts as (
select
  a.bin_28d as cohort_dt,
  b.bin_28d as invoice_dt
from all_28d_bins a
cross join all_28d_bins b
where b.bin_28d >= a.bin_28d
)

, period_stats_by_cohort as (
select
  c.owner_cohort_bin_28d,
  i.bin_28d,
  sum(1) as workspaces
from enriched_invoices i
join owner_cohort c
  on c.owner_id = i.owner_id
group by 1,2
)
```

```
-- , filled_up_periods as (
select
  cohort_dt,
  invoice_dt,
  row_number() over(partition by cohort_dt order by invoice_dt) as cohort_age_28d,
  coalesce(workspaces,0) as workspaces,
  1- coalesce(workspaces,0) ::float
          / first_value(coalesce(workspaces,0)) over(partition by cohort_dt order by invoice_dt) as
workspace_churn
from all_28d_bins_cohorts c
left join period_stats_by_cohort s
  on c.invoice_dt = s.bin_28d
  and c.cohort_dt = s.owner_cohort_bin_28d
where invoice_dt < '2018-12-30 00:00:00' -- last and partial invoice 28d period
order by 1,2
```